Vulnerability Scan Report

HostedScan Security



Table of Contents

- Report Summary
- Passive Web Application Vulnerabilities
- Active Web Application Vulnerabilities
- SSL/TLS Security
- Network Vulnerabilities
- Open TCP Ports
- Open UDP Ports

Report Summary

This report contains information on all of the risks found from your vulnerability scans. Each risk is assigned a threat level (high, medium, or low).

Total Risks

Total number of risks found by severity.

2 High

18 Medium 9 Low 0 Accepted

Passive Web Application Vulnerabilities

The OWASP ZAP passive web application scan crawls the pages of a web application. It inspects the web pages as well as the requests and responses sent between the server. The passive scan checks for vulnerabilities such as cross-domain misconfigurations, insecure cookies, vulnerable js dependencies, and more.

Total Risks

Total number of risks found by the passive web application vulnerability scan.



5 Medium 4 Low

0 Accepted

Risks Summary

Summary of detected risks.

Threat Level	Title		Accepted Count
MEDIUM	Absence of Anti-CSRF Tokens	1	0
MEDIUM	Cross-Domain Misconfiguration	1	0
MEDIUM	<u>Vulnerable JS Library</u>	1	0
MEDIUM	Missing Anti-clickjacking Header	1	0
MEDIUM	Referer Exposes Session ID	1	0
LOW	X-Content-Type-Options Header Missing	1	0
LOW	Cookie Without Secure Flag	1	0
LOW	Cookie No HttpOnly Flag	1	0
LOW	Cookie without SameSite Attribute	1	0

Risks Per Target Summary

Breakdown of risk counts for each target.

Target	High	Medium	Low	Accepted
Project https://edistrict.kerala.gov.in	0	5	4	0

Full Risk Details

Detailed information about each risk found by the scan.

Absence of Anti-CSRF Tokens



Name

Absence of Anti-CSRF Tokens

No Anti-CSRF tokens were found in a HTML submission form.

A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.

Description

CSRF attacks are effective in a number of situations, including:

- * The victim has an active session on the target site.
- * The victim is authenticated via HTTP auth on the target site.
- * The victim is on the same local network as the target site.

CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Phase: Implementation

Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.

Phase: Architecture and Design

Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).

Solution

Note that this can be bypassed using XSS.

Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.

Note that this can be bypassed using XSS.

Use the ESAPI Session Management control.

This control includes a component for CSRF.

Do not use the GET method for any request that triggers a state change.

Phase: Implementation

Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

Reference

http://projects.webappsec.org/Cross-Site-Request-Forgery

http://cwe.mitre.org/data/definitions/352.html

CWE Id

352

WASC Id

9

Other Information

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "btnSubmit" "fromDate" "hiddenEvent" "rdoID" "rdoName" "toDate" "txtFrom" "txtTo"].

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Cross-Domain Misconfiguration

MEDIUM

Name

Cross-Domain Misconfiguration

Description

Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server

Solution

Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance).

Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.

Reference

https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy

CWE Id

264

WASC Id

14

Other Information

The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Vulnerable JS Library

MEDIUM

N	а	m	16	9

Vulnerable JS Library

Description

The identified library bootstrap, version 3.3.7 is vulnerable.

Solution

Please upgrade to the latest version of bootstrap.

Reference

https://github.com/twbs/bootstrap/issues/28236

Reference

https://github.com/twbs/bootstrap/issues/20184

https://github.com/advisories/GHSA-4p24-vmcr-4gqj

CWE Id

829

WASC Id

-1

CVE-2019-8331

CVE-2018-14041

Other Information

CVE-2018-14040

CVE-2018-14042

CVE-2016-10735

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Missing Anti-clickjacking Header

MEDIUM

Name Missing Anti-clickjacking Header

Description

The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.

Solution

 $\label{thm:content-Security-Policy} Modern \ Web \ browsers \ support \ the \ Content-Security-Policy \ and \ X-Frame-Options \ HTTP \ headers. \ Ensure one of them is set on all web pages returned by your site/app.$

0...

If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

Reference

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

CWE Id

1021

WASC Id

15

Other Information

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Referer Exposes Session ID

MEDIUM

Name

Referer Exposes Session ID

Description

A hyperlink pointing to another host name was found. As session ID URL rewrite is used, it may be disclosed in referer header to external hosts.

Solution

This is a risk if the session ID is sensitive and the hyperlink refers to an external or third party host. For secure content, put session ID in secured session cookie.

Reference

http://seclists.org/lists/webappsec/2002/Oct-Dec/0111.html

CWE Id

200

13

WASC Id

Other Information

Vulnerable Target	Accepted	Notes

Project

https://edistrict.kerala.gov.in

X-Content-Type-Options Header Missing



Name

X-Content-Type-Options Header Missing

Description

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

ation

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Reference

 $\underline{\text{http://msdn.microsoft.com/en-us/library/ie/gg622941\%28v=vs.85\%29.aspx}}$

https://owasp.org/www-community/Security_Headers

CWE Id

693

WASC Id

15

Other Information

This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type.

At "High" threshold this scan rule will not alert on client or server error responses.

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Cookie Without Secure Flag



Name	Cookie Without Secure Flag
Description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.
Reference	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
CWE Id	614
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes	
Project			
https://edistrict.kerala.gov.in			

Cookie No HttpOnly Flag

Vulnerable Target

LOW

Notes

Name	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly
CWE Id	1004
WASC Id	13
Other Information	

Accepted

Cookie without SameSite Attribute



Name	Cookie without SameSite Attribute
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	1275
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes	
Project			
https://edistrict.kerala.gov.in			

Active Web Application Vulnerabilities

The OWASP ZAP active web application scan crawls the pages of a web application. It scans for all of the passive scan checks and additionally makes requests and submits forms to actively test for more vulnerabilities. The active scan checks for vulnerabilities such as SQL injection, remote command execution, XSS, and more.

Total Risks

Total number of risks found by the active web application vulnerability scan.



6 Medium 4 Low 0 Accepted

Risks Summary

Summary of detected risks.

Threat Level	Title		Accepted Count
HIGH	Cross Site Scripting (Reflected)	1	0
MEDIUM	Absence of Anti-CSRF Tokens	1	0
MEDIUM	.htaccess Information Leak	1	0
MEDIUM	Cross-Domain Misconfiguration	1	0
MEDIUM	<u>Vulnerable JS Library</u>	1	0
MEDIUM	Missing Anti-clickjacking Header	1	0
MEDIUM	Session ID in URL Rewrite	1	0
LOW	Cookie No HttpOnly Flag	1	0
LOW	X-Content-Type-Options Header Missing	1	0
LOW	Cookie Without Secure Flag	1	0
LOW	Cookie without SameSite Attribute	1	0

Risks Per Target Summary

Breakdown of risk counts for each target.

Target	High	Medium	Low	Accepted
Project https://edistrict.kerala.gov.in	1	6	4	0

Full Risk Details

Detailed information about each risk found by the scan.

Cross Site Scripting (Reflected)



Name

Cross Site Scripting (Reflected)

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise.

Description

There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based.

Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash.

Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code.

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket.

Phases: Implementation; Architecture and Design

Understand the context in which your data will be used and the encoding that will be expected. This is especially important when transmitting data between different components, or when generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data representations to determine the required encoding strategies.

For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters.

Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Solution

If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated.

Phase: Implementation

For every web page that is generated, use and specify a character encoding such as ISO-8859-1 or UTF-8. When an encoding is not specified, the web browser may choose a different encoding by guessing which encoding is actually being used by the web page. This can cause the web browser to treat certain sequences as special, opening up the client to subtle XSS attacks. See CWE-116 for more mitigations related to encoding/escaping.

To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as more recent versions of Internet Explorer and Firefox), this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie. This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHTTPRequest and other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in which the HttpOnly flag is set.

Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a deny list). However, deny lists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.

When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue."

Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere.

Reference

http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html

CWE Id

79

WASC Id

8

Other Information

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Absence of Anti-CSRF Tokens



Name

Absence of Anti-CSRF Tokens

No Anti-CSRF tokens were found in a HTML submission form.

A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.

Description

CSRF attacks are effective in a number of situations, including:

- * The victim has an active session on the target site.
- * The victim is authenticated via HTTP auth on the target site.
- * The victim is on the same local network as the target site.

CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the sameorigin policy.

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Phase: Implementation

Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.

Phase: Architecture and Design

Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).

Solution

Note that this can be bypassed using XSS.

Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.

Note that this can be bypassed using XSS.

Use the ESAPI Session Management control.

This control includes a component for CSRF.

Do not use the GET method for any request that triggers a state change.

Phase: Implementation

Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

Reference

http://projects.webappsec.org/Cross-Site-Request-Forgery

http://cwe.mitre.org/data/definitions/352.html

CWE Id

352

WASC Id

9

Other Information

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "btnSubmit" "fromDate" "hiddenEvent" "rdoID" "rdoName" "toDate" "txtFrom" "txtTo"].

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

.htaccess Information Leak

MEDIUM

Name

.htaccess Information Leak

Description

htaccess files can be used to alter the configuration of the Apache Web Server software to enable/disable additional functionality and features that the Apache Web Server software has to offer.

Solution

Ensure the .htaccess file is not accessible.

Reference

http://www.htaccess-guide.com/

CWE Id

94

WASC Id

14

Other Information

Vulnerable Target

Accepted

Notes

Project

https://edistrict.kerala.gov.in

Cross-Domain Misconfiguration

MEDIUM

Name

Cross-Domain Misconfiguration

Description

Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server

Solution

Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance).

Solution

Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.

Reference

https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy

CWE Id

264

WASC Id

14

Other Information The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Vulnerable JS Library



Name	Vulnerable JS Library
Description	The identified library bootstrap, version 3.3.7 is vulnerable.
Solution	Please upgrade to the latest version of bootstrap.
Reference	https://github.com/twbs/bootstrap/issues/28236 https://github.com/twbs/bootstrap/issues/20184 https://github.com/advisories/GHSA-4p24-vmcr-4gqj
CWE Id	829
WASC Id	-1
Other Information	CVE-2019-8331 CVE-2018-14041 CVE-2018-14040 CVE-2018-14042 CVE-2016-10735

Vulnerable Target	Accepted	Notes

Project https://edistrict.kerala.gov.in

Missing Anti-clickjacking Header

MEDIUM

Name	Missing Anti-clickjacking Header
Description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
Solution	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

CWE Id 1021
WASC Id 15
Other Information

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Session ID in URL Rewrite

MEDIUM

Name Session ID in URL Rewrite

Description

URL rewrite is used to track user session ID. The session ID may be disclosed via cross-site referer header. In addition, the session ID might be stored in browser history or server logs.

Solution

For secure content, put session ID in a cookie. To be even more secure consider using a combination of cookie and URL rewrite.

Reference

http://seclists.org/lists/webappsec/2002/Oct-Dec/0111.html

CWE Id

200

WASC Id

13

Other Information

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

Cookie No HttpOnly Flag

LOW

Name Cookie No HttpOnly Flag

Description

A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

Solution

Ensure that the HttpOnly flag is set for all cookies.

Reference

https://owasp.org/www-community/HttpOnly

CWE Id 1004
WASC Id 13
Other Information

Vulnerable Target Accepted Notes

Project

https://edistrict.kerala.gov.in

X-Content-Type-Options Header Missing

LOW

Name

X-Content-Type-Options Header Missing

Description

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Reference

http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx

https://owasp.org/www-community/Security_Headers

CWE Id

693

WASC Id

15

Other Information

This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type.

At "High" threshold this scan rule will not alert on client or server error responses.

Vulnerable Target Accepted Notes

Proiect

https://edistrict.kerala.gov.in

Cookie Without Secure Flag

LOW

Name	Cookie Without Secure Flag
Description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.
Reference	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
CWE Id	614
WASC Id	13
Other Information	

 Vulnerable Target
 Accepted
 Notes

 Project https://edistrict.kerala.gov.in

Cookie without SameSite Attribute

LOW

Name	Cookie without SameSite Attribute
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	1275
WASC Id	13
Other Information	

 Vulnerable Target
 Accepted
 Notes

 Project https://edistrict.kerala.gov.in

SSL/TLS Security

 $The \,SSLyze \,security \,scan \,checks \,for \,misconfigured \,SSL/TLS \,certificates, \,expired \,certificates, \,weak \,ciphers, \,and \,SSL/TLS \,vulnerabilities \,such \,as \,Heartbleed.$

Total Risks

Total number of risks found by the SSL/TLS security scan.







0 Accepted

Network Vulnerabilities

The OpenVAS network vulnerability scan tests servers and network connected devices for over 50,000 vulnerabilities.

Total Risks

Total number of risks found by the network vulnerability scan.





1 Low 0 Accepted

Risks Summary

Summary of detected risks.

Threat Level	Title	CVSS Score		Accepted Count
HIGH	Apache Axis2 Document Type Declaration Processing Security Vulnerability	7.5	1	0
MEDIUM	HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8	1	0
MEDIUM	SSL/TLS: Report Weak Cipher Suites	5.0	1	0
MEDIUM	Missing `httpOnly` Cookie Attribute	5.0	1	0
MEDIUM	MacOS X Finder '.DS_Store' Information Disclosure	5.3	1	0
MEDIUM	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	4.3	1	0
MEDIUM	<u>Apache Axis2 <= 1.6.2 Multiple Vulnerabilities</u>	6.4	1	0
MEDIUM	Apache Axis2 engagingglobally Cross-Site Scripting Vulnerability	4.3	1	0
Low	TCP timestamps	2.6	1	0

Risks Per Target Summary

Breakdown of risk counts for each target.

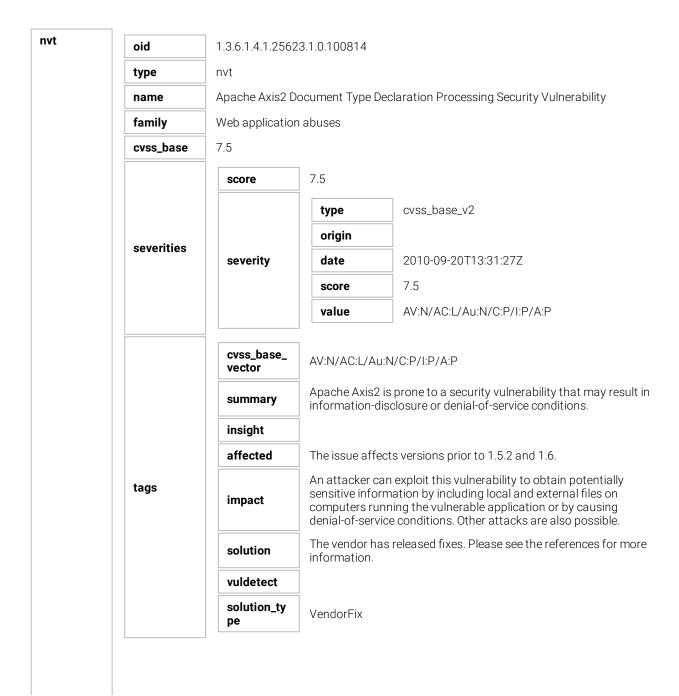
Target	High	Medium	Low	Accepted
Project https://edistrict.kerala.gov.in	1	7	1	0

Full Risk Details

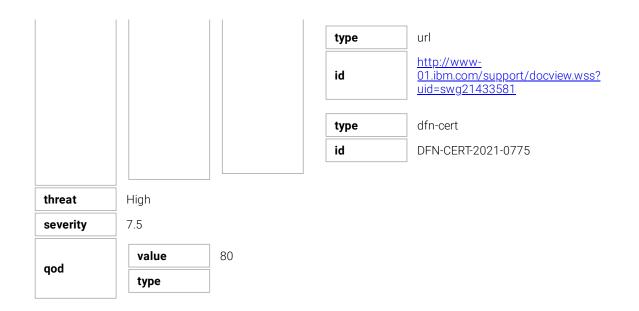
Detailed information about each risk found by the scan.

Apache Axis2 Document Type Declaration Processing Security Vulnerability





solution		information.		
type		VendorFix		
refs	ref	*	01/0	
		type	cve	
		id	CVE-2010-1632	
		type	url	
		id	http://www.securityfocus.com/bid/40976	
		type	url	
		id	http://ws.apache.org/axis2/	
		type	url	
		туре		
		id	http://geronimo.apache.org/2010/07/21/apache-geronimo-v216-released.html	
		type	url	
		id	http://www- 01.ibm.com/support/docview.wss?	
		lu lu	uid=swg27019456	
		type	url	
		id	https://issues.apache.org/jira/browse/AX IS2-4450	
		type	url	
		id	https://svn.apache.org/repos/asf/axis/axis/java/core/security/CVE-2010-	
			<u>1632.pdf</u>	
		type	url	
		id	http://www- 01.ibm.com/support/docview.wss?	
		lu lu	rs=180&uid=swg24027020	
		type	url	
		:	http://www-	
		id	01.ibm.com/support/docview.wss? rs=180&uid=swg24027019	
		type	url	
		id	http://www.ibm.com/support/docview.ws s?uid=swg24027503	
		type	url	
		id	http://www.ibm.com/support/docview.ws	



Vulnerable Target Accepted Notes

Project https://edistrict.kerala.gov.in

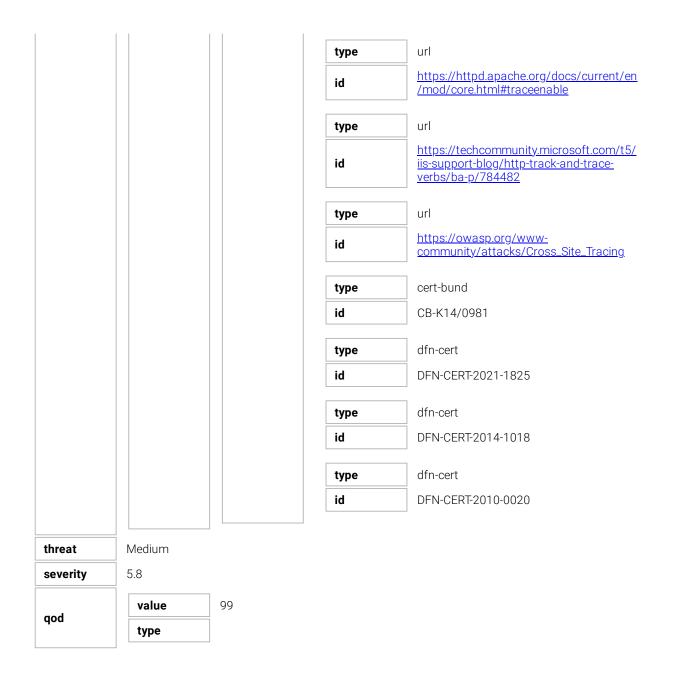
HTTP Debugging Methods (TRACE/TRACK) Enabled

MEDIUM



cvss_base_ vector		AV:N/AC:M/Au	u:N/C:P/I:P/A:N		
		methods. TRA	eb server supports the TRACE and/or TRACK CE and TRACK are HTTP methods which are use rver connections.		
	insight	It has been shown that web servers supporting this methods are subject to cross-site-scripting attacks, dubbed XST for Cross-Site-Tracing, when used in conjunction with various weaknesses in browsers.			
tags	affected	Web servers w	rith enabled TRACE and/or TRACK methods.		
3 -	impact	An attacker may use this flaw to trick your legitimate web users t give him their credentials.			
	solution	configuration.	Disable the TRACE and TRACK methods in your web server configuration. Please see the manual of your web server or the references for more information.		
	vuldetect	Checks if HTTI and can be us	P methods such as TRACE and TRACK are enable ed.		
	solution_ty pe	Mitigation			
solution		configuration.	Disable the TRACE and TRACK methods in your web server configuration. Please see the manual of your web server or the references for more information.		
	type	Mitigation			
refs		<u>-</u>]			
	ref	type	cve		
		id	CVE-2003-1567		
		type	cve		
		id	CVE-2004-2320		
		Id	GVE-2004-2320		
		type	cve		
		id	CVE-2004-2763		
		type	cve		
		id	CVE-2005-3398		
		I U	CVL 2000 3090		
		type	cve		
		id	CVE-2006-4683		
		type	cve		
		id	CVE-2007-3008		
		type	cve		
		id	CVE-2008-7253		
		type	cve		

type	cve
id	CVE-2010-0386
type	cve
id	CVE-2012-2223
type	cve
id	CVE-2014-7883
type	url
id	http://www.kb.cert.org/vuls/id/288308
type	url
id	http://www.securityfocus.com/bid/11604
type	url
id	http://www.securityfocus.com/bid/15222
type	url
id	http://www.securityfocus.com/bid/19915
type	url
id	http://www.securityfocus.com/bid/24456
type	url
id	http://www.securityfocus.com/bid/33374
type	url
id	http://www.securityfocus.com/bid/36956
type	url
id	http://www.securityfocus.com/bid/36990
type	url
id	http://www.securityfocus.com/bid/37995
type	url
id	http://www.securityfocus.com/bid/9506
type	url
id	http://www.securityfocus.com/bid/9561
type	uri
id	http://www.kb.cert.org/vuls/id/867593



 Vulnerable Target
 Accepted
 Notes

 Project
 https://edistrict.kerala.gov.in

SSL/TLS: Report Weak Cipher Suites

MEDIUM

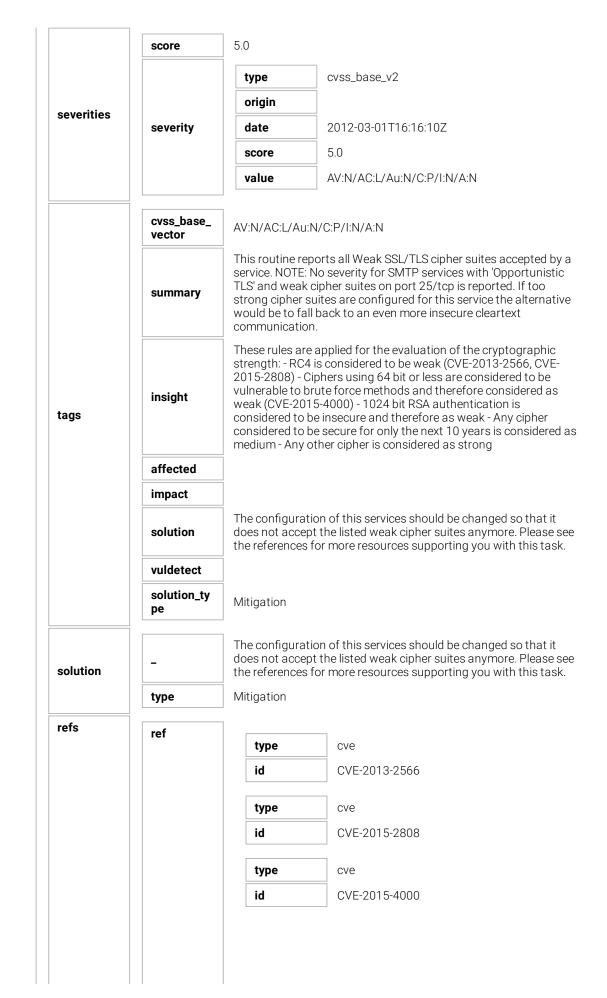
 oid
 1.3.6.1.4.1.25623.1.0.103440

 type
 nvt

 name
 SSL/TLS: Report Weak Cipher Suites

 family
 SSL and TLS

 cvss_base
 5.0



type	url
id	https://www.bsi.bund.de/SharedDocs/Vrnmeldungen/DE/CB/warnmeldung_cb-k16-1465_update_6.html
type	url
id	https://bettercrypto.org/
type	url
id	https://mozilla.github.io/server-side- tls/ssl-config-generator/
type	cert-bund
id	CB-K21/0067
type	cert-bund
id	CB-K19/0812
type	cert-bund
id	CB-K17/1750
type	cert-bund
id	CB-K16/1593
type	cert-bund
id	CB-K16/1552
type	cert-bund
id	CB-K16/1102
type	cert-bund
id	CB-K16/0617
type	cert-bund
id	CB-K16/0599
type	cert-bund
id	CB-K16/0168
type	cert-bund
id	CB-K16/0121
type	cert-bund
id	CB-K16/0090

	_
type	cert-bund
id	CB-K16/0030
type	cert-bund
id	CB-K15/1751
type	cert-bund
id	CB-K15/1591
type	cert-bund
id	CB-K15/1550
type	cert-bund
id	CB-K15/1517
type	cert-bund
id	CB-K15/1514
type	cert-bund
id	CB-K15/1464
type	cert-bund
id	CB-K15/1442
type	cert-bund
id	CB-K15/1334
typo	cert-bund
id	CB-K15/1269
-]
type	cert-bund
id	CB-K15/1136
type	cert-bund
id	CB-K15/1090
type	cert-bund
id	CB-K15/1059
type	cert-bund
id	CB-K15/1022
type	cert-bund
· · · · · · · · · · · · · · · · · · ·	-

type	cert-bund
id	CB-K15/0986
type	cert-bund
id	CB-K15/0964
type	cert-bund
id	CB-K15/0962
type	cert-bund
id	CB-K15/0932
type	cert-bund
id	CB-K15/0927
type	cert-bund
id	CB-K15/0926
type	cert-bund
id	CB-K15/0907
type	cert-bund
id	CB-K15/0901
type	cert-bund
id	CB-K15/0896
type	cert-bund
id	CB-K15/0889
type	cert-bund
id	CB-K15/0877
type	cert-bund
id	CB-K15/0850
type	cert-bund
id	CB-K15/0849
type	cert-bund
id	CB-K15/0834
type	cert-bund
id	CB-K15/0827
Iu	OB 1(10,002)

type	cert-bund
id	CB-K15/0802
type	cert-bund
id	CB-K15/0764
type	cert-bund
id	CB-K15/0733
type	cert-bund
id	CB-K15/0667
type	cert-bund
id	CB-K14/0935
type	cert-bund
id	CB-K13/0942
type	dfn-cert
id	DFN-CERT-2021-0775
type	dfn-cert
id	DFN-CERT-2020-1561
type	dfn-cert
id	DFN-CERT-2020-1276
type	dfn-cert
id	DFN-CERT-2017-1821
type	dfn-cert
id	DFN-CERT-2016-1692
type	dfn-cert
id	DFN-CERT-2016-1648
type	dfn-cert
id	DFN-CERT-2016-1168
type	dfn-cert
id	DFN-CERT-2016-0665
type	dfn-cert
id	DFN-CERT-2016-0642
iu	DIN OLNI 2010-0042

type	dfn-cert
id	DFN-CERT-2016-0184
type	dfn-cert
id	DFN-CERT-2016-0135
type	dfn-cert
id	DFN-CERT-2016-0101
type	dfn-cert
id	DFN-CERT-2016-003
type	dfn-cert
id	DFN-CERT-2015-1853
type	dfn-cert
id	DFN-CERT-2015-1679
type	dfn-cert
id	DFN-CERT-2015-1632
type	dfn-cert
id	DFN-CERT-2015-1608
type	dfn-cert
id	DFN-CERT-2015-1542
type	dfn-cert
id	DFN-CERT-2015-1518
type	dfn-cert
id	DFN-CERT-2015-1406
type	dfn-cert
id	DFN-CERT-2015-134
type	dfn-cert
id	DFN-CERT-2015-1194
type	dfn-cert
id	DFN-CERT-2015-1144
type	dfn-cert
id	DFN-CERT-2015-1113

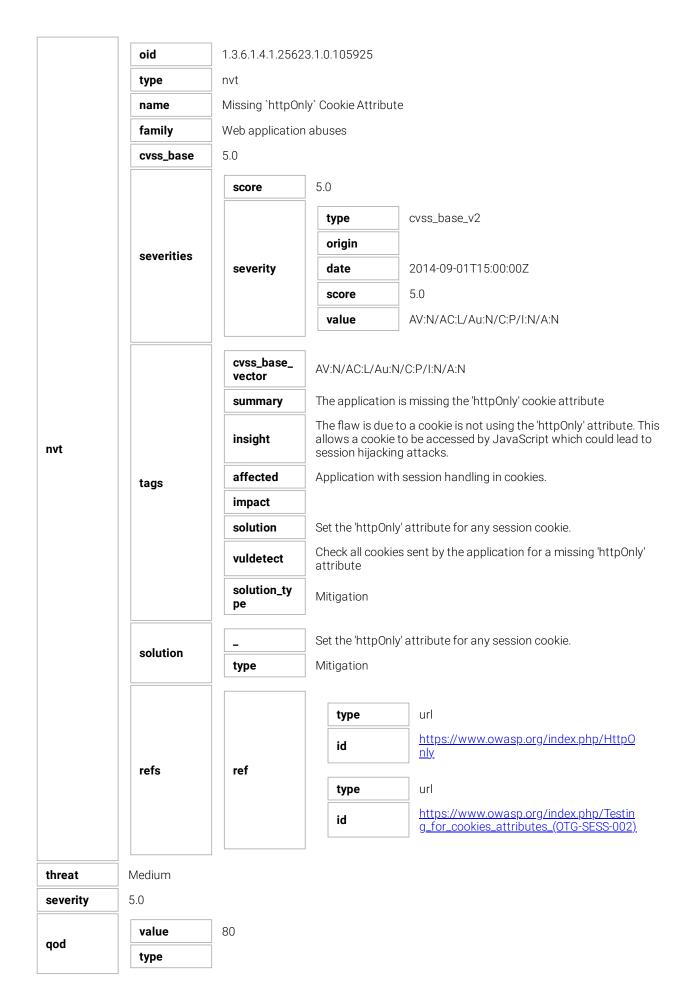
type	dfn-cert
id	DFN-CERT-2015-1078
type	dfn-cert
id	DFN-CERT-2015-1067
type	dfn-cert
id	DFN-CERT-2015-1038
type	dfn-cert
id	DFN-CERT-2015-1016
type	dfn-cert
id	DFN-CERT-2015-1012
type	dfn-cert
id	DFN-CERT-2015-0980
type	dfn-cert
id	DFN-CERT-2015-0977
type	dfn-cert
id	DFN-CERT-2015-0976
type	dfn-cert
id	DFN-CERT-2015-0960
type	dfn-cert
id	DFN-CERT-2015-0956
type	dfn-cert
id	DFN-CERT-2015-0944
type	dfn-cert
id	DFN-CERT-2015-0937
type	dfn-cert
id	DFN-CERT-2015-0925
type	dfn-cert
id	DFN-CERT-2015-0884
type	dfn-cert
id	DFN-CERT-2015-0881



 Vulnerable Target
 Accepted
 Notes

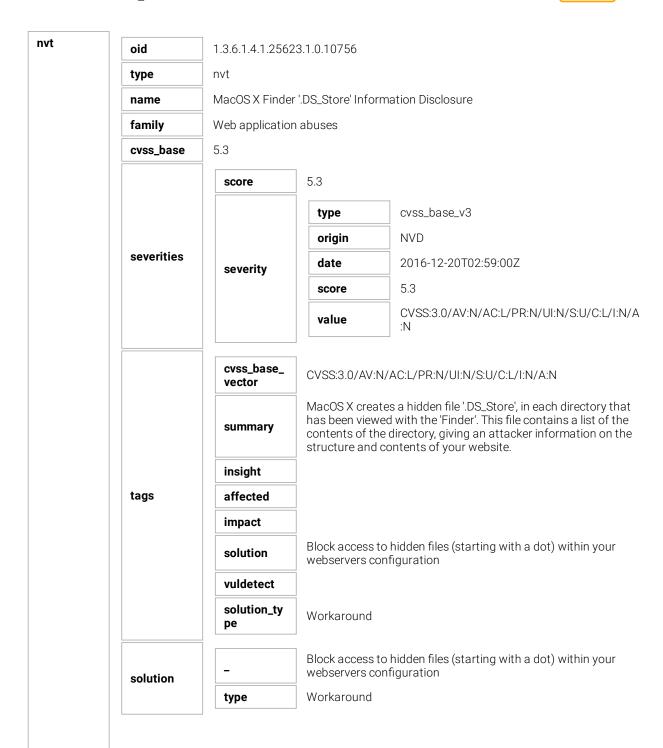
 Project https://edistrict.kerala.gov.in

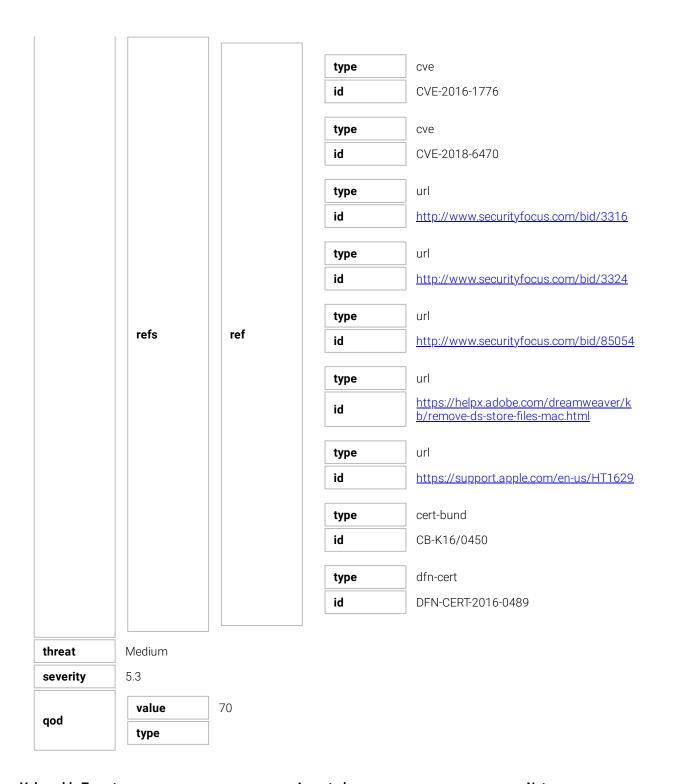
Missing `httpOnly` Cookie Attribute



Project https://edistrict.kerala.gov.in

MacOS X Finder '.DS_Store' Information Disclosure





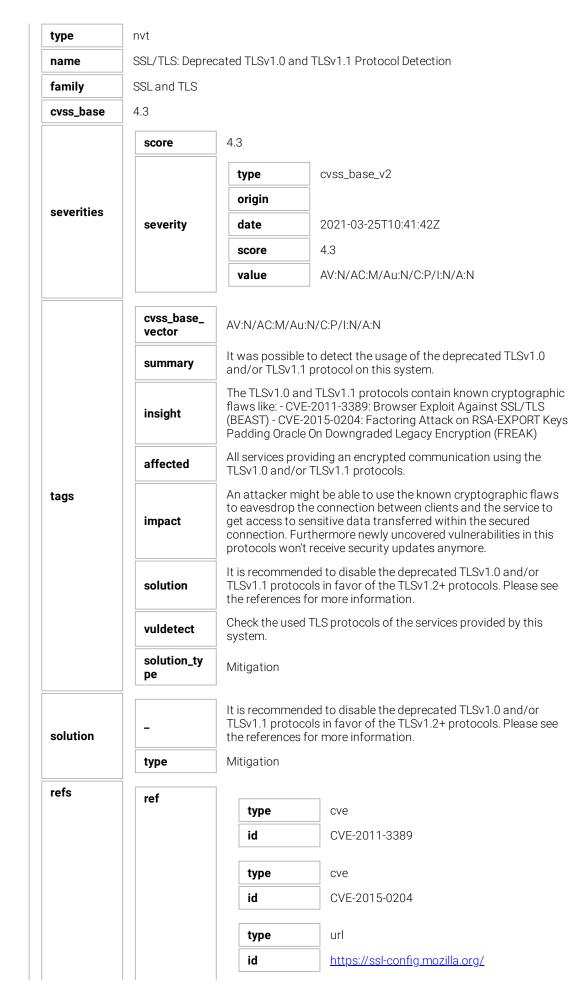
 Vulnerable Target
 Accepted
 Notes

 Project https://edistrict.kerala.gov.in

SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection

MEDIUM

nvt oid 1.3.6.1.4.1.25623.1.0.117274



type	url
id	https://bettercrypto.org/
type	url
id	https://datatracker.ietf.org/doc/rfc8996/
type	url
id	https://vnhacker.blogspot.com/2011/09/ beast.html
type	url
id	https://web.archive.org/web/202011080 5603/https://censys.io/blog/freak
type	url
id	https://www.enisa.europa.eu/publication/algorithms-key-size-and-parameters-report-2014
type	cert-bund
id	CB-K18/0799
type	cert-bund
id	CB-K16/1289
type	cert-bund
id	CB-K16/1096
type	cert-bund
id	CB-K15/1751
type	cert-bund
id	CB-K15/1266
type	cert-bund
id	CB-K15/0850
type	cert-bund
id	CB-K15/0764
type	cert-bund
id	CB-K15/0720
type	cert-bund
id	CB-K15/0548

type	cert-bund
id	CB-K15/0526
type	cert-bund
id	CB-K15/0509
type	cert-bund
id	CB-K15/0493
type	cert-bund
id	CB-K15/0384
type	cert-bund
id	CB-K15/0365
type	cert-bund
id	CB-K15/0364
type	cert-bund
id	CB-K15/0302
type	cert-bund
id	CB-K15/0192
type	cert-bund
id	CB-K15/0079
type	cert-bund
id	CB-K15/0016
4 1.00 A	
type	cert-bund
id	CB-K14/1342
type	cert-bund
id	CB-K14/0231
type	cert-bund
id	CB-K13/0845
tune	oort hund
type : a	cert-bund
id	CB-K13/0796
type	cert-bund
id	CB-K13/0790

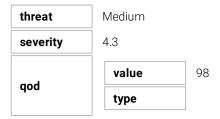
type	dfn-cert
id	DFN-CERT-2020-0177
type	dfn-cert
id	DFN-CERT-2020-011
type	dfn-cert
id	DFN-CERT-2019-0068
type	dfn-cert
id	DFN-CERT-2018-144
type	dfn-cert
id	DFN-CERT-2018-1408
type	dfn-cert
id	DFN-CERT-2016-1372
type	dfn-cert
id	DFN-CERT-2016-1164
type	dfn-cert
id	DFN-CERT-2016-0388
type	dfn-cert
id	DFN-CERT-2015-1853
type	dfn-cert
id	DFN-CERT-2015-1332
type	dfn-cert
id	DFN-CERT-2015-0884
type	dfn-cert
id	DFN-CERT-2015-0800
type	dfn-cert
id	DFN-CERT-2015-0758
type	dfn-cert
id	DFN-CERT-2015-0567
type	dfn-cert
	DFN-CERT-2015-0544

type	dfn-cert
id	DFN-CERT-2015-0530
type	dfn-cert
id	DFN-CERT-2015-0396
type	dfn-cert
id	DFN-CERT-2015-037
type	dfn-cert
id	DFN-CERT-2015-0374
type	dfn-cert
id	DFN-CERT-2015-030
type	dfn-cert
id	DFN-CERT-2015-0199
type	dfn-cert
id	DFN-CERT-2015-0079
type	dfn-cert
id	DFN-CERT-2015-0021
type	dfn-cert
id	DFN-CERT-2014-1414
type	dfn-cert
id	DFN-CERT-2013-1847
type	dfn-cert
id	DFN-CERT-2013-1792
type	dfn-cert
id	DFN-CERT-2012-1979
type	dfn-cert
id	DFN-CERT-2012-1829
type	dfn-cert
id	DFN-CERT-2012-1530
type	dfn-cert
id	DFN-CERT-2012-1380

type	dfn-cert
id	DFN-CERT-2012-1377
type	dfn-cert
id	DFN-CERT-2012-1292
type	dfn-cert
id	DFN-CERT-2012-1214
type	dfn-cert
id	DFN-CERT-2012-1213
type	dfn-cert
id	DFN-CERT-2012-1180
type	dfn-cert
id	DFN-CERT-2012-1156
type	dfn-cert
id	DFN-CERT-2012-115
type	dfn-cert
id	DFN-CERT-2012-1039
type	dfn-cert
id	DFN-CERT-2012-0956
type	dfn-cert
id	DFN-CERT-2012-0908
type	dfn-cert
id	DFN-CERT-2012-0868
type	dfn-cert
id	DFN-CERT-2012-0867
type	dfn-cert
id	DFN-CERT-2012-0848
type	dfn-cert
id	DFN-CERT-2012-0838
type	dfn-cert

type	dfn-cert
id	DFN-CERT-2012-0722
type	dfn-cert
id	DFN-CERT-2012-0638
type	dfn-cert
id	DFN-CERT-2012-0627
type	dfn-cert
id	DFN-CERT-2012-0451
type	dfn-cert
id	DFN-CERT-2012-0418
type	dfn-cert
id	DFN-CERT-2012-0354
type	dfn-cert
id	DFN-CERT-2012-0234
type	dfn-cert
id	DFN-CERT-2012-022
type	dfn-cert
id	DFN-CERT-2012-017
type	dfn-cert
id	DFN-CERT-2012-0170
type	dfn-cert
id	DFN-CERT-2012-0146
type	dfn-cert
id	DFN-CERT-2012-0142
type	dfn-cert
id	DFN-CERT-2012-0126
type	dfn-cert
id	DFN-CERT-2012-0123
type	dfn-cert
id	DFN-CERT-2012-0095

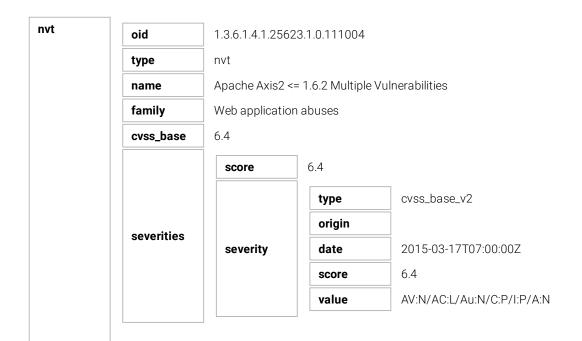
DFN-CERT-2012-0051
dfn-cert
DFN-CERT-2012-0047
dfn-cert
DFN-CERT-2012-0021
dfn-cert
DFN-CERT-2011-1953
dfn-cert
DFN-CERT-2011-1946
dfn-cert
DFN-CERT-2011-1844
dfn-cert
DFN-CERT-2011-1826
dfn-cert
DFN-CERT-2011-1774
dfn-cert
DFN-CERT-2011-1743
dfn-cert
DFN-CERT-2011-1738
dfn-cert
DFN-CERT-2011-1706
dfn-cert
DFN-CERT-2011-1628
dfn-cert
DFN-CERT-2011-1627
dfn-cert
DFN-CERT-2011-1619
dfn-cert
uiii ocit



 Vulnerable Target
 Accepted
 Notes

 Project https://edistrict.kerala.gov.in

Apache Axis2 <= 1.6.2 Multiple Vulnerabilities



	cvss_base_ vector	AV:N/AC:L/Au:	N/C:P/I:P/A:N	
	summary	Apache Axis2 i	s prone to multiple vulnerabilities.	
	insight	The following flaws exist: - a security-bypass vulnerability becauthe application fails to properly validate SSL certificates from the server a security vulnerability involving XML signature wrappir		
	affected	The issue affec	ets versions up to 1.6.2.	
tags	impact	Successfully exploiting these issues allows attackers to: - performan-in-the-middle attacks or impersonate trusted servers, which will aid in further attacks may allow unauthenticated attacker to construct specially crafted messages that can be successfully verified and contain arbitrary content. This may aid in further attacks.		
	solution	No known solution was made available for at least one year sin the disclosure of this vulnerability. Likely none will be provided anymore. General solution options are to upgrade to a newer release, disable respective features, remove the product or replathe product by another one.		
	vuldetect	Checks if a vul	nerable version is present on the target host.	
	solution_ty pe	WillNotFix		
solution	-	the disclosure anymore. Gene	ition was made available for at least one year sir of this vulnerability. Likely none will be provided eral solution options are to upgrade to a newer e respective features, remove the product or repl another one.	
	type	WillNotFix		
		ιype	cve	
		id	cve CVE-2012-5785	
		id	CVE-2012-5785	
		id type id	CVE-2012-5785	
		id type	CVE-2012-5785 cve CVE-2012-4418	
refs	ref	id type id type id	CVE-2012-5785 cve CVE-2012-4418 cve CVE-2012-5351	
refs	ref	id type id type	CVE-2012-5785 cve CVE-2012-4418 cve	
refs	ref	id type id type id type id	CVE-2012-5785 cve CVE-2012-4418 cve CVE-2012-5351 url http://www.securityfocus.com/bid/56408	
refs	ref	id type id type id type	CVE-2012-5785 cve CVE-2012-4418 cve CVE-2012-5351 url	
refs	ref	id type id type id type id type id	cve CVE-2012-4418 cve CVE-2012-5351 url http://www.securityfocus.com/bid/56408 url http://www.securityfocus.com/bid/55508	
refs	ref	id type id type id type id type	CVE-2012-5785 cve CVE-2012-4418 cve CVE-2012-5351 url http://www.securityfocus.com/bid/56408 url	

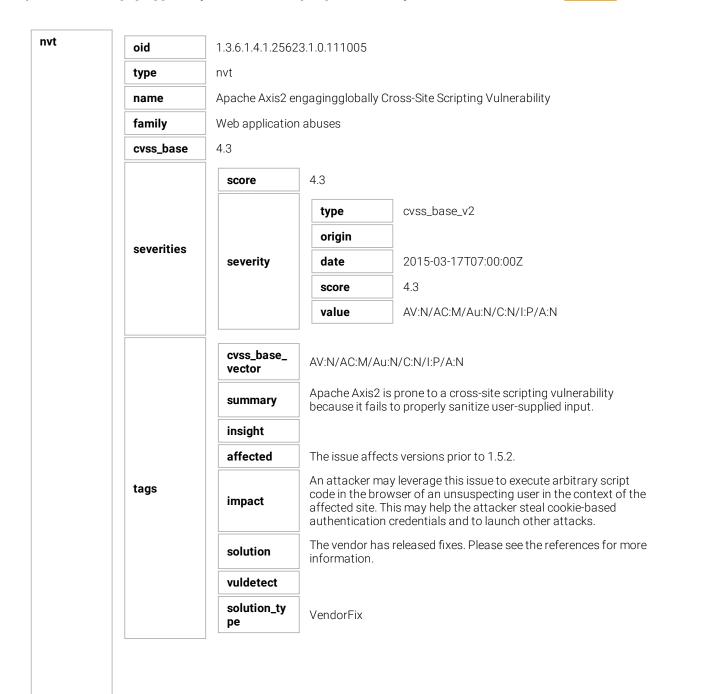
threat



 Vulnerable Target
 Accepted
 Notes

 Project
 https://edistrict.kerala.gov.in

Apache Axis2 engagingglobally Cross-Site Scripting Vulnerability



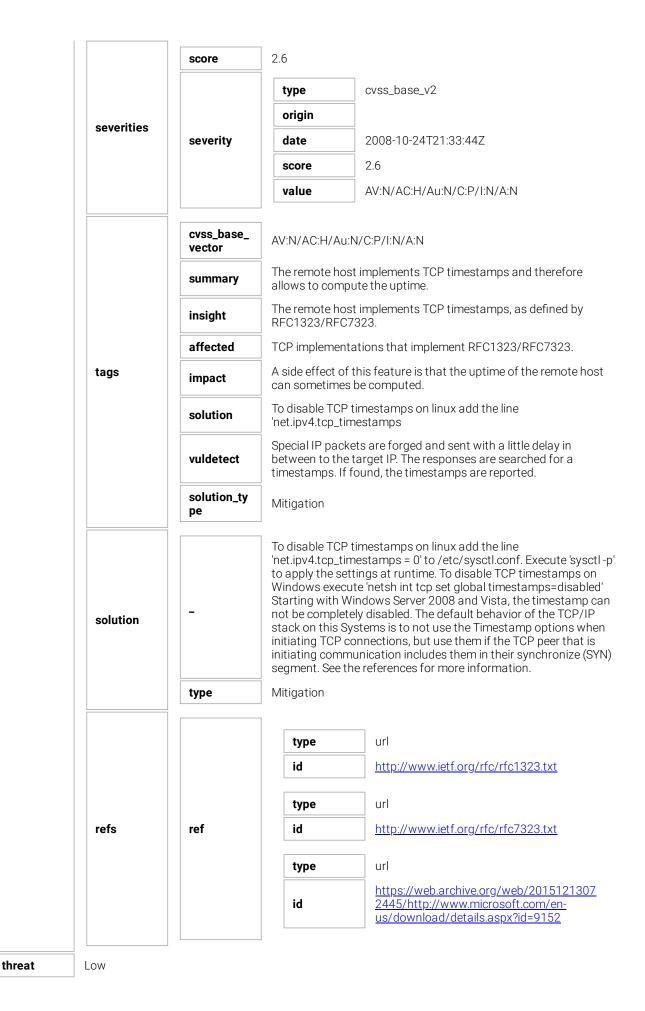


Vulnerable Target Accepted Notes Project https://edistrict.kerala.gov.in

TCP timestamps

LOW







Vulnerable Target Accepted Notes

Project https://edistrict.kerala.gov.in

Open TCP Ports

The NMAP TCP port scan discovers open ports on with a complete scan of ports 0 to 65535.

Total Risks

Total number of risks found by the TCP port scan.





0 Low 0 Accepted

Open UDP Ports

The NMAP UDP port scan discovers open ports of common UDP services.

Total Risks

Total number of risks found by the UDP port scan.



0 Medium 0 Low 0 Accepted